



SnapManager® for SAP

Best Practices

SAP Competence Center, Network Appliance
August 2007 | TR-3582

Abstract

This document provides customers and partners with the best practices for deploying SnapManager for SAP.



Table of Contents

1	Introduction	3
2	Landscape Planning	5
2.1	Storage Layout for SAP Systems and SnapManager for SAP	5
2.2	Repository Layout	5
2.3	Disaster Recovery	7
3	Installation	7
3.1	Installing the SnapManager for SAP GUI	7
3.2	Installing the Repository	8
3.3	Registering Systems in the Repository	8
3.4	Creating Profiles	9
4	Using SnapManager for SAP	10
4.1	Backing Up Systems	10
4.2	Restoring Systems	13
4.3	Cloning Systems	15
4.4	Maintenance Tasks	16
	Appendix A – Sample Fibre Channel Storage Setup	18
	Appendix B – Sample Cloning Specification – XML Format	19
	Appendix C – Cloning Post Processing Script – smsap_clone_post_processing.sh	21
	Appendix D – Clone Post Processing Template – insert_sapuser_pw.sql	24
	Appendix E – Clone Post Processing Template – smsap_post_processing.tpl	25
	Appendix F – Temp Tablespace File Add – add_temp_file.sql	26
	Additional References	27

1 Introduction

Scope

This document is intended to provide customers and partners with the best practices for utilizing NetApp SnapManager for SAP in SAP Business Suite Applications using Oracle® Databases. Primary consideration has been given to addressing the planning, installation, and use of the product by IT personnel responsible for managing SAP solutions. Recommendations are generic and specific to neither a particular SAP application nor the size and scope of SAP implementation. This guide assumes a basic understanding of the technology and operation of NetApp and SAP products.

Business challenges

Corporations deploying SAP solutions today are under great pressure to reduce TCO, accelerate ROI, and increase productivity and availability of their SAP landscapes through infrastructure simplification. Restructuring activities, mergers and acquisitions, and constantly changing market conditions often result in the creation of new ERP landscapes based on the SAP NetWeaver™ technology platform. SAP NetWeaver permits more flexible adoption and integration of new business processes and scenarios. Timely access to data and the ability to analyze it not only becomes possible, it becomes a requirement for corporations to keep pace with change.

IT challenges

A typical production SAP landscape today consists of several different SAP systems. Successful operation and management of these production systems depends on careful attention to the numerous nonproduction instances that are required to support it.

SAP has long encouraged their customers to maintain separate development and quality assurance instances for each production instance. In practice, it is not uncommon for such a three system landscape to be expanded to include separate systems supporting functions such as a technical sandbox and training. Driven by standard processes for development and testing within a corporation, it is also typical to have multiple development instances as well as more than one system utilized for quality assurance, additional testing, or perhaps a final staging system prior to releasing changes into production.

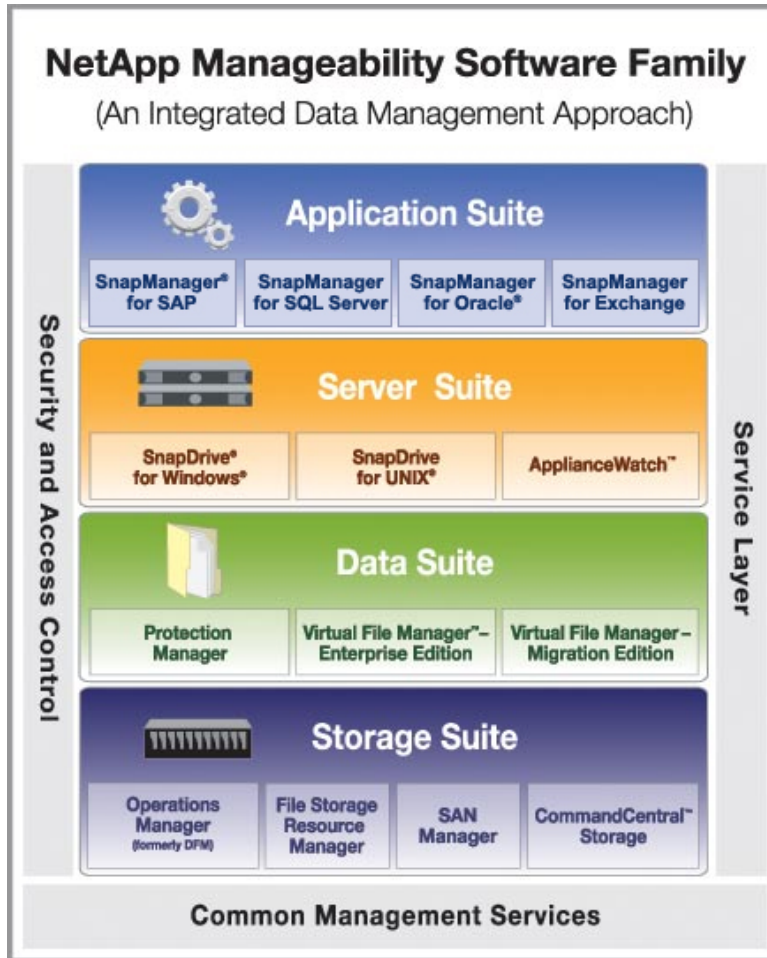
Adding to the challenge of maintaining these databases and the servers needed to drive them is the fact that each of these instances is going to have differing performance, scalability, availability and uptime profiles. These profiles can also fluctuate depending on the phases of a project's implementation and whether the project is focused on an existing SAP implementation or on a brand new one.

In summary, for each instance of SAP running in production, there can be as few as two and perhaps five or more instances supporting it. Deploying three SAP applications, like R/3, CRM, and BW (as each of these applications requires its own database instance) can easily result in IT departments having to account for 15 or more SAP instances in total. All of these instances then need to be backed up, copied, or cloned to support test schedules or to create a reference instance for new projects, and factored into a disaster recovery plan.

If the IT infrastructure supporting SAP applications is inflexible, difficult to operate or manage, or has a high cost of ownership barriers develop within the IT department, which can negatively impact the ability of business owners to deploy new and improved business processes.

Network Appliance SnapManager for SAP

A key to the Network Appliance storage simplification strategy is enabling application administrators to take advantage of the capabilities of our storage solutions without requiring extensive training and experience with storage. To that end, we have developed a suite of tools for managing applications that are integrated with both the application and the storage.



SnapManager for SAP is designed to address the need to manage SAP systems running on Oracle databases. First and foremost, SnapManager for SAP is designed to quickly and efficiently backup SAP systems. SnapManager for SAP is a SAP-certified backup solution, using the SAP BACKINT interface to backup and restore SAP Oracle databases utilizing the unique NetApp Snapshot[™] functionality. By providing a robust command line interface and intuitive graphical user interface, SnapManager for SAP removes the complexity from these operations. By taking advantage of the Data ONTAP[®], FlexClone[®], and LUN clone features, SnapManager for SAP provides a mechanism for creating SAP system copies in a few minutes without the costly overhead of allocating large amounts of storage. SnapManager for SAP can remove the risk and complexity from daily system management activities.

2 Landscape Planning

2.1 Storage Layout for SAP Systems and SnapManager for SAP

For environments where NFS is used to connect the storage to the host, there are no special requirements for the storage layout. Please review the document [SAP on UNIX and Oracle with NFS and Network Appliance Storage](#) for recommendations on how to install your SAP system using NetApp storage and the NFS protocol.

For environments using fibre channel protocol, there is an additional requirement for the storage layout. SnapManager for SAP requires that each file system containing data that you wish to backup with SnapManager for SAP reside in its own disk group or volume group. This is easily accomplished using SnapDrive® for UNIX®. An example of the SnapDrive commands required to provision storage for an SAP system is included in Appendix A. For additional information, refer to the document [SAP on UNIX and Oracle with FCP and Network Appliance Storage](#).

2.2 Repository Layout

SnapManager for SAP uses a repository to store information about each system that it backs up. This repository is stored in an Oracle database. Part of the installation process involves loading the SnapManager for SAP schema into a previously installed Oracle database. However, determining which repository will hold the data for which SAP systems is an important part of the planning process.

How many repositories to install

As with any database application, having a consistent backup in case of a failure of the infrastructure is important and SnapManager for SAP is no different. Because the repository resides in an Oracle database, being able to backup and restore the repository is critical. SnapManager for SAP cannot backup itself. The target database for backup cannot be the same database where the database profile resides. Therefore, the best practice of using SnapManager for SAP to backup the repository requires that you install at least two repositories. You can install more than two repositories, but the management of numerous repositories can be complex. The total number of repositories depends on how you choose to organize the systems into the repositories.

How to distribute systems in the repositories

There are several different ways to organize systems in repositories. We will describe several scenarios for distribution of systems. The first method for distributing systems would be based on the application type. You would install a separate repository for every SAP application type that you have. For example if you had ECC, CRM, BI, and Portal applications, you would have four repositories. Each repository would have a profile for the systems of that application type. Your development, quality assurance, and production systems would all be backed up from the same repository. This is useful because it keeps like systems together and makes cloning of systems convenient. However, because of the number of different applications in SAP landscapes, you may have a large number of repositories to manage. Also, if you choose to implement another SAP application, for example APO or SRM, you will need to install another SnapManager for SAP repository. Because of the need to be able to backup and restore production instances, each of those repositories would need to be on a server with continuous availability, which could be expensive. If security is a consideration, having production systems in every repository could be an issue. If you want to restrict those users who have access to backup, restore, or clone production systems, they would need access to multiple repositories which means additional work for the people managing the access.

Another option for distributing systems in repositories is to divide them among repositories based on system type (i.e., development, quality assurance, production, training, etc.). This will limit the number of repositories to the number of different types of systems that you have employed, and it reduces the need to manage access to all of the repositories as all production systems would be together in the same repository. Also, if you choose to install another SAP application, it's only necessary to register the system in the proper repository rather than install a new repository. Because all of the production systems would reside in the same repository, it's only necessary to provide high availability for that one repository rather than every repository.

A variation of the previous distribution would be to put the production systems in a single repository and all of the other nonproduction systems in a single repository. This has the same benefits of the previous example along with the added bonus of the number of repositories required to manage the backups for the landscape being reduced to only two.

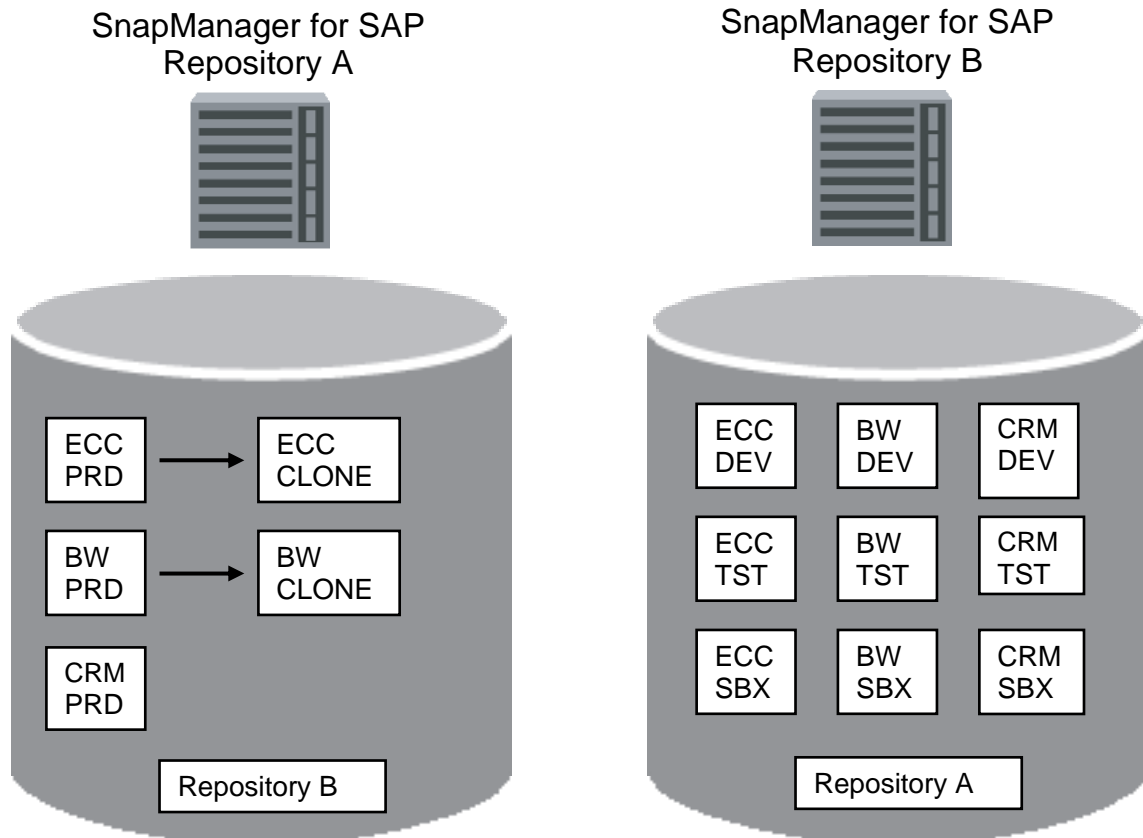


Figure 1) Sample Repository Layout

How big to make the repository

The size of the repository will determine how many backup records it can contain. Our testing has shown that the size of a single backup record in the repository is approximately 100kb. At this size, even a repository of 100MB could hold approximately 1000 backup records. This should be sufficient for the majority of customer environments. For a customer environment with eight systems, that would be 250 backups each.

High Availability

SnapManager for SAP is an important part of your overall SAP landscape. It enables fast, efficient backup and restore operations, the ability to recover SAP systems in a fraction of the time that a traditional tape backup architecture would require, and the freedom to create SAP system copies in a matter of minutes. However, with that additional functionality comes the need to manage the SnapManager for SAP repository much like you would any other database in your landscape. If the server hosting SnapManager for SAP is down, you may be unable to backup, restore, and clone your systems, which can impact your business.

As a result, it is necessary to provide a high availability environment for your SnapManager for SAP repository. Providing high availability for SnapManager for SAP is no different than for any other Oracle database

environment. If you have followed the best practice of installing more than one repository, it is possible for a single repository server to go down and still be able to perform backups by simply registering the systems contained in the disabled repository in the one that is still active. The limitation is that you would not be able to restore any of the backups that are recorded in the disabled repository. Another option is to install the repository on a server cluster, following the best practices for creating a high availability Oracle environment. This would allow you to maintain normal operations if one server should fail.

2.3 Disaster Recovery

SnapManager for SAP is not a replacement for backups to secondary storage or tape. SnapManager for SAP creates backups of SAP systems using Snapshot copies on a NetApp storage system. This allows for fast back-ups and restores, but does not provide security in the event of a failure on the storage system itself. While these failures are rare, there is the potential that they could occur and so it is a best practice to always maintain a backup of your systems in a secondary location or media.

This is also true in the case of a failure of the storage where the SnapManager for SAP repository is located. The only solution would be to restore from a backup. A failure on the storage system could result in loss of SnapManager for SAP data related to backups and clones. As a result, a best practice is to always backup the SnapManager for SAP repository to a secondary location to prevent data loss in the case of a storage system failure.

Backing up your data to a secondary location can be accomplished in several ways. The simplest method for creating a secondary backup would be to perform a backup to tape. This is a common method for creating backups for disaster recovery purposes. Some special considerations need to be made when performing tape backups and using SnapManager for SAP. If you use backint-compatible tape backup software, it will be necessary to replace the NetApp backint file with the backint file provided by the tape backup solution vendor. Once the backup is complete, you will have to revert back to the NetApp SnapManager backint file. This process could be easily scripted to avoid errors.

Another method for creating a tape backup would be to use a dedicated media server to backup the database by attaching a clone of the system. You could use SnapManager for SAP to create a clone from a Snapshot copy of the database and then mount that clone to the media server. The media server would then run the backup software and allow you to make a tape copy of the database for disaster recovery purposes. This provides the benefit of having a tape backup without any impact to the system and also without having to move backint files. There is some additional cost related to having the dedicated media server.

Finally, NetApp provides software for disaster recovery scenarios. SnapMirror® is a Data ONTAP software feature that allows you to create copies of data on a secondary storage system. It is a highly efficient IP-based method for migrating data to a disaster recovery location. If desired, a backup to tape can be taken from the secondary location using a Snapshot copy of the data on the secondary storage system.

Currently there is no SnapVault® integration for SnapManager for SAP v1.0, but it is planned in a later release.

3 Installation

3.1 Installing the SnapManager for SAP GUI

The SnapManager for SAP GUI is not required to perform any SnapManager functions. All activities can be performed using the SnapManager for SAP command line. However, the GUI greatly simplifies these tasks.

The SnapManager for SAP GUI can be installed on a host running the Windows® XP operating system. There are no special requirements for installing the GUI. Refer to the *SnapManager for SAP v1.0 Installation and Administration Guide* on the [NOW](#) (NetApp on the Web) site for instructions on installing the GUI.

Because there is no one-to-one requirement for the SnapManager GUI and repositories, you can potentially have many SnapManager GUIs administering profiles in many repositories. The only limitation is that for a user to add a

repository to their GUI, they need to know where the repository is installed and the necessary passwords for connecting to it.

3.2 Installing the Repository

Before any systems can be managed using SnapManager for SAP, an Oracle database has to be configured to store the profiles for each system. The process is straightforward and can be found in the *SnapManager for SAP v 1.0 Installation and Administration Guide* on the NOW site.

Where to install the repository

Because the SnapManager for SAP repository resides in an Oracle database, a new database installation is required. It would be possible to use an existing SAP Oracle database to store the repository; however, this is not recommended for several reasons. First, SAP discourages storing data other than SAP application data in a SAP database. Second, the Oracle versions and patch levels supported by SnapManager for SAP may be different than those required for SAP applications. This could lead to the solution being unsupported and possibly to a loss of data or backups. Finally, if the repository is stored in a SAP database and the SAP system is undergoing maintenance like an upgrade, kernel patch, or support packages, none of the systems in that repository could be backed up.

As a result, it is a best practice to install a separate database instance to store the SnapManager for SAP repository.

Installing an Oracle database for the repository

There are no special requirements for the database that contains the SnapManager for SAP repository. The only requirement is that there is sufficient space for the data. The amount of space required for the repository is based on the number of systems contained in the repository and the number of backups retained for each system.

Creating the repository schema from the SnapManager GUI

The easiest way to create the repository schema is from the SnapManager GUI. Once the repository database is installed and running, you can then start them from the SnapManager GUI. From the GUI, select "Operations → Repository → Create New Repository..." This will open the Create Repository wizard and lead you through the creation of the repository.

3.3 Registering Systems in the Repository

Once the repository and SnapManager GUI are installed, you can begin to register the systems in the repository. The registering of the systems happens on the host that you want to back up. You need to be sure that the SnapManager for SAP server and database are running before you can perform these steps. Make sure that SnapManager for SAP is running on the system to be backed up.

First, it is necessary to give SnapManager for SAP access to the source host to execute the backup commands. As the ora<sid> or <sid>adm user, issue the following command:

```
smsap credential set -host -name <source system host> -password <ora<sid> password>
```

Next, grant the ora<sid> userid on the source system access to the repository with the following command. This will set the password for the source system to the repository so that it can be authenticated without having to enter the password for every command.

```
smsap credential set -repository -host <repository host> -dbname smsap  
-port <repository listener port number> -login -username <repository userid> -password <repository  
password>
```

Then, give the source ora<sid> userid visibility to the profiles in the repository with the following command. The password was set in the previous step, so there is no need to send the password.


```
smsap profile sync -repository -host <repository host> -dbname <repository database name> -port  
<repository listener port> -login -username <repository userid>
```

Finally, set the password for the profile so that the source system ora<sid> userid does not have to authenticate with a password for actions on that profile, (see example below). The profile needs to be created before you issue this command. You can create the profile via the command line or from the SnapManager GUI. See section 3.4 Creating Profiles to learn how to create a profile. Or, refer to the *SnapManager for SAP v 1.0 Installation and Administration Guide*.

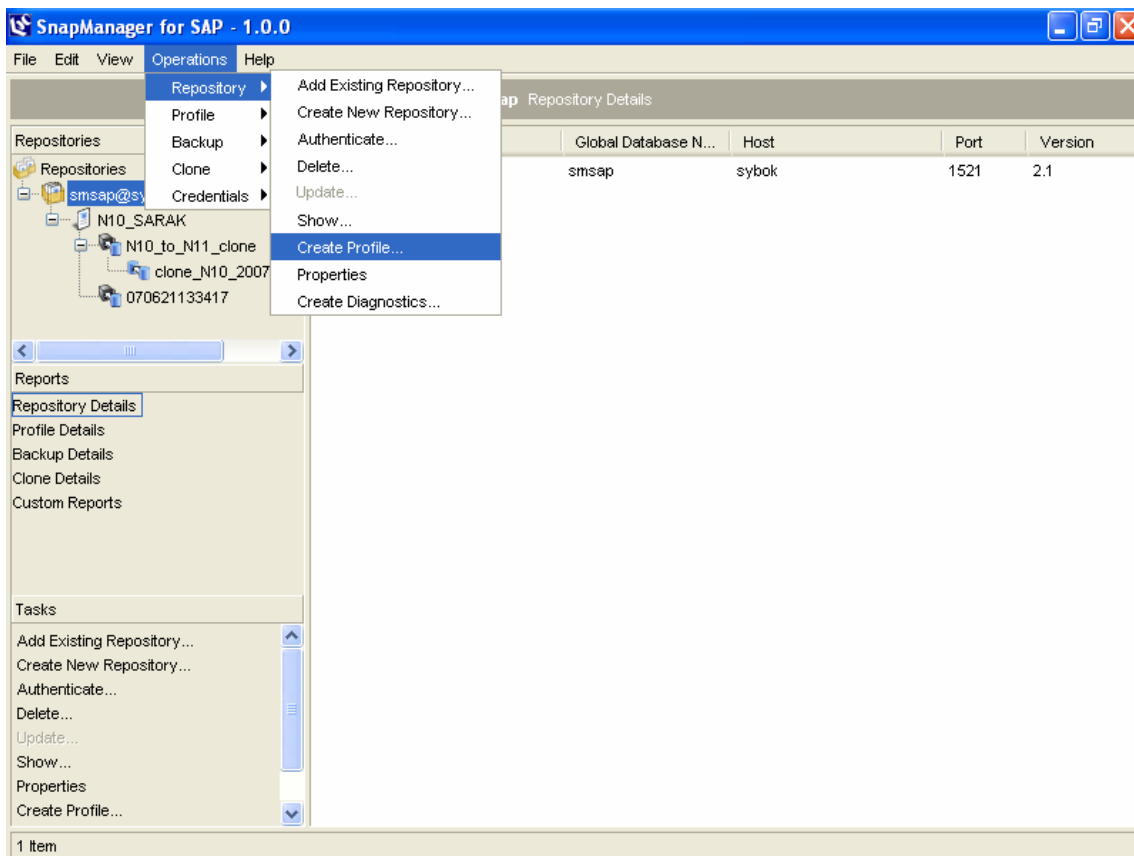
```
smsap credential set -profile -name <profile name> -password <profile password>
```

These steps need to be completed for every system that you want to manage with SnapManager for SAP.

3.4 Creating Profiles

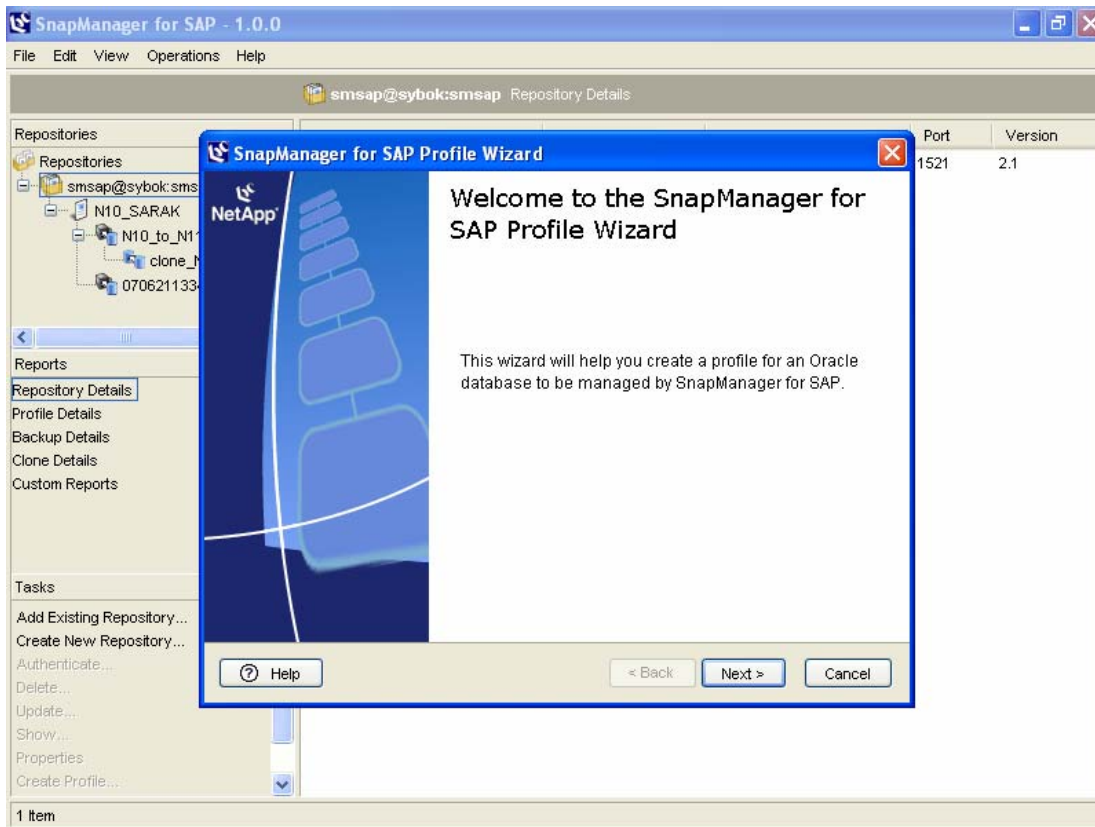
In order to be able to perform operations on a source database, a profile has to be created that registers the database in the SnapManager for SAP repository.

This is most easily accomplished using the SnapManager GUI. From the GUI select “Operations → Repository → Create Profile...”



You can also right-click on the repository and select “Create Profile” from the pull down menu.

This will open the Profile wizard to guide you through the steps for creating the profile.



When creating the profile, you will need to provide a userid and password for the database that has sysdba privileges. You can choose to create a new database user for this purpose or use an existing Oracle standard userid. Rather than increase the number of userids to administer, we chose to use the standard Oracle user "sys." While we recommend using a standard Oracle userid to perform these activities, we also recognize that some customers may feel uncomfortable doing this because of security reasons. Each customer must balance their own need for simplicity with security. Any userid with sysdba privileges on the database will work for performing SnapManager for SAP activities. This allows our customers the flexibility they need to determine what is best for their environments.

4 Using SnapManager for SAP

4.1 Backing Up Systems

Using SnapManager for SAP, you have several choices for creating backups of your SAP systems. You can use BRBACKUP to create a SAP-certified backup of your SAP system using the backint interface. You can schedule SAP-certified backups to occur automatically using the database scheduler transaction DB13. Or you can create a backup via the SnapManager for SAP GUI to be used as the source for a system clone.

Backing up systems using BRBACKUP

Using the BR*Tools program BRBACKUP to backup your SAP using SnapManager for SAP is the same as using it to backup your SAP system to a tape drive. Once you have installed SnapManager for SAP and created the necessary profiles, simply configure BRBACKUP to create a "utility" backup and then run BRBACKUP with the appropriate options. You can also specify that BRBACKUP make a backup to a "utility" by using the "-d" option in the BRBACKUP command. For example, to take an online backup, issue the following command as user ora<sid>

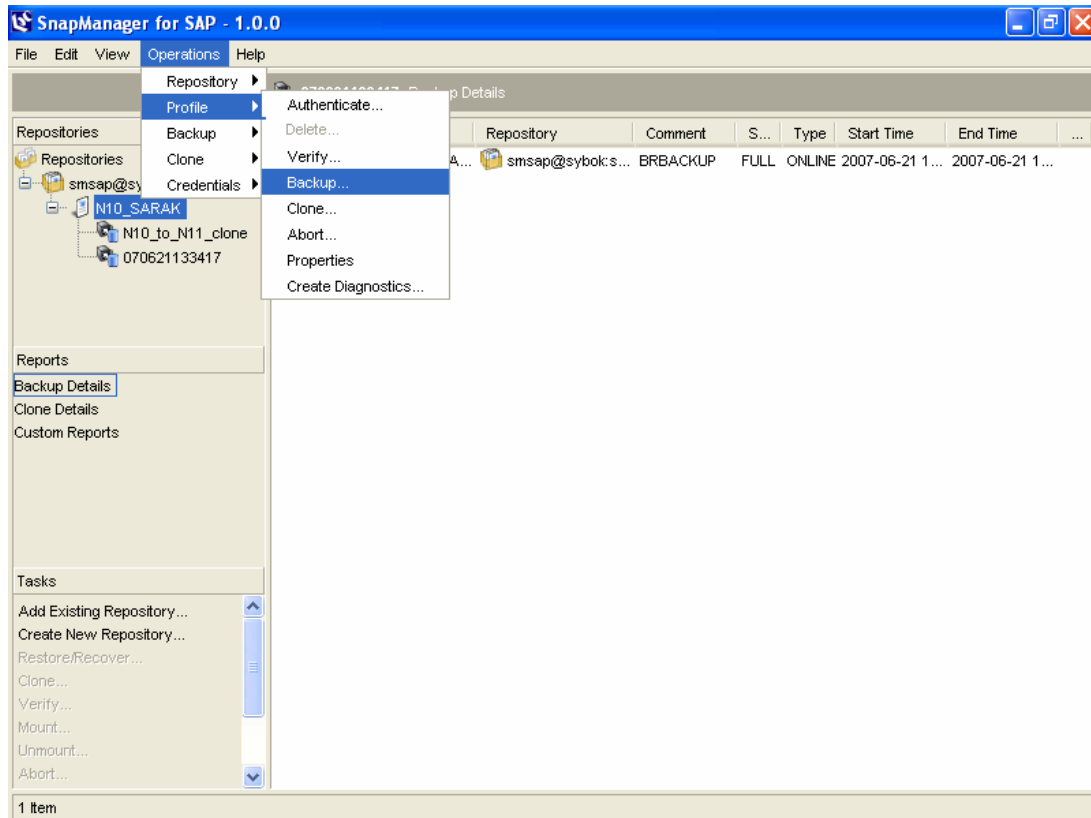
```
brbackup -c -d util_file_online -m all -t online -u system/<password>
```

To configure BRBACKUP to create “utility” backups by default, set “backup_dev_type = util_file” in the <ORACLE_HOME>/dbs/init<SID>.sap file. This will eliminate the need to use the “-d” option in the BRBACKUP command every time.

Backing up systems using the SnapManager for SAP GUI

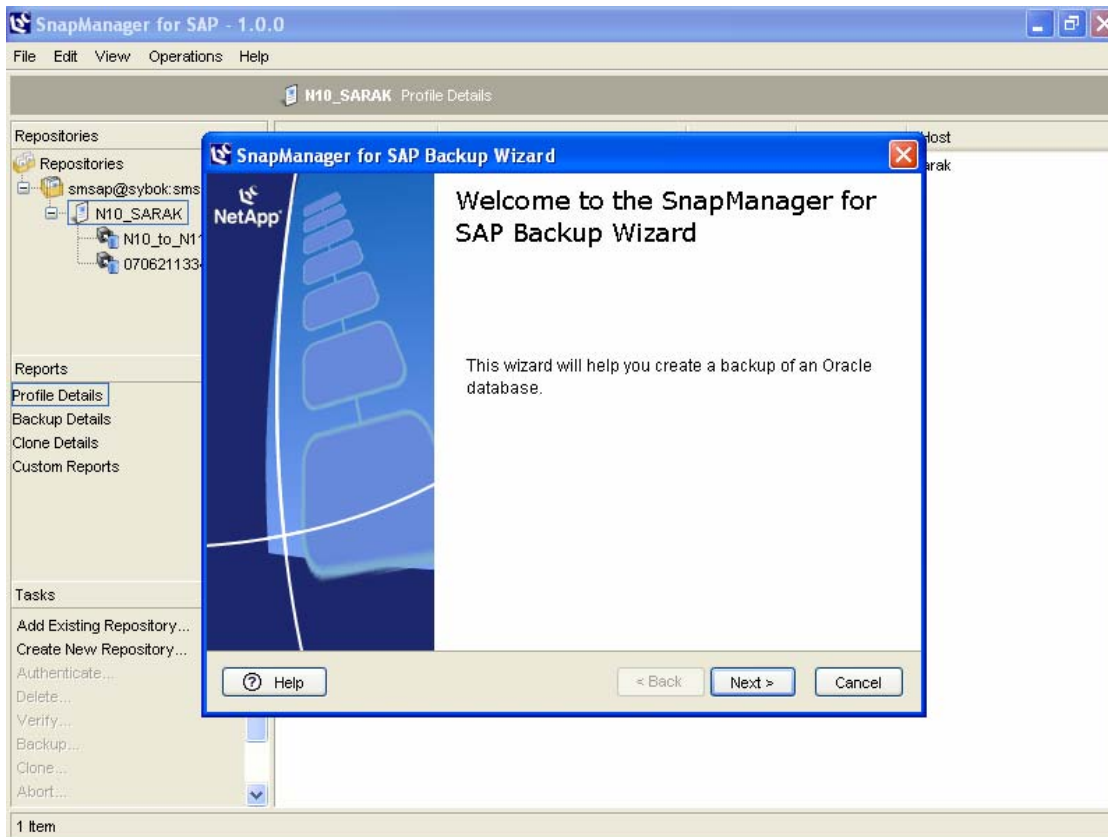
You can also use SnapManager for SAP to backup your SAP systems without BRBACKUP. However, this does not create a SAP-certified backup and should only be used when creating a backup to be used as a source for a clone operation.

To create the backup, open the SnapManager for SAP GUI and select the profile you want to backup, and follow the menu path “Operations → Profile → Backup” to open the Backup wizard.



You can also choose the profile that you would like to create and right-click. Then choose “Backup.”

Follow the steps in the SnapManager for SAP Backup wizard to create the backup.



Backing up systems using the SAP Database Planning Calendar transaction DB13

Because SnapManager for SAP is integrated with SAP BR*Tools, backups can be scheduled using the SAP Database Planning Calendar.

There are some changes that need to be made to the configuration of the system before backups can be run from the SAP Database Planning Calendar. The changes required can be found by logging on to the [NetApp Knowledge Base](#) and reviewing the article titled *Scheduling SnapManager for SAP backups via SAP transaction DB13* by searching for article number kb29946.

Once these changes have been made, scheduling a backup using SnapManager for SAP is the same as scheduling a backup with any third party software. Simply schedule the backup as you would normally. Refer to the [SAP Help Portal](#) for detailed instructions for using the Database Planning Calendar.

Backing up archive logs using BRARCHIVE

Using the BR*Tools program BRARCHIVE, it is possible to backup archive logs to Snapshot copies. However, depending on the storage protocol you are using, this may not be the most efficient method of creating backups of the archive logs. Also, as with the database data files, it is always good to have a secondary backup of all data on separate media or storage system.

When using NFS to connect NetApp storage, using BRARCHIVE and deleting the archive logs will not free up space in the file system or the NetApp volume. The Snapshot copy of the data in the file system is kept until the Snapshot is deleted. As a result, using SnapManager for SAP to backup archive logs in an NFS environment is not an efficient use of resources. In an NFS environment, it is more efficient to backup the archive logs to another NFS-mounted file system and then move them to tape from that location. This serves the purpose of having them available for fast restore from disk, but also frees space in the Oracle archive file system. Another option is to back

them up directly to tape. This will increase the time necessary to recover the database if a roll-forward operation is required, but it will also free up the space in the Oracle archive file system. In order to backup the archive logs using third-party backup software, it will be necessary to replace the NetApp backint file with the third-party backint file before performing the backup, and then replace the NetApp file when you want to use SnapManager to backup the database.

When using FCP to connect NetApp storage for your SAP systems, using BRARCHIVE and deleting the archive logs will free space in the filesystem. However, it will not free space in the NetApp volume that contains the LUNS for the device where the filesystem resides. As a result, it may be necessary to closely monitor the volumes where the LUN for the Oracle archive directory resides to be sure that it does not run out of space. It is always good to have a secondary backup of all data on a separate media or storage system.

4.2 Restoring Systems

Restoring backups using BRRESTORE

The process for restoring systems using SnapManager for SAP and BRRESTORE is the same as restoring with any other utility-based backup solution. It is necessary to determine the log file name for the backup you want to restore and then issue the BRRESTORE command as user ora<sid>. For example, to restore the online backup with the logfile bdvlfxx.anf, issue the following command as user ora<sid>:

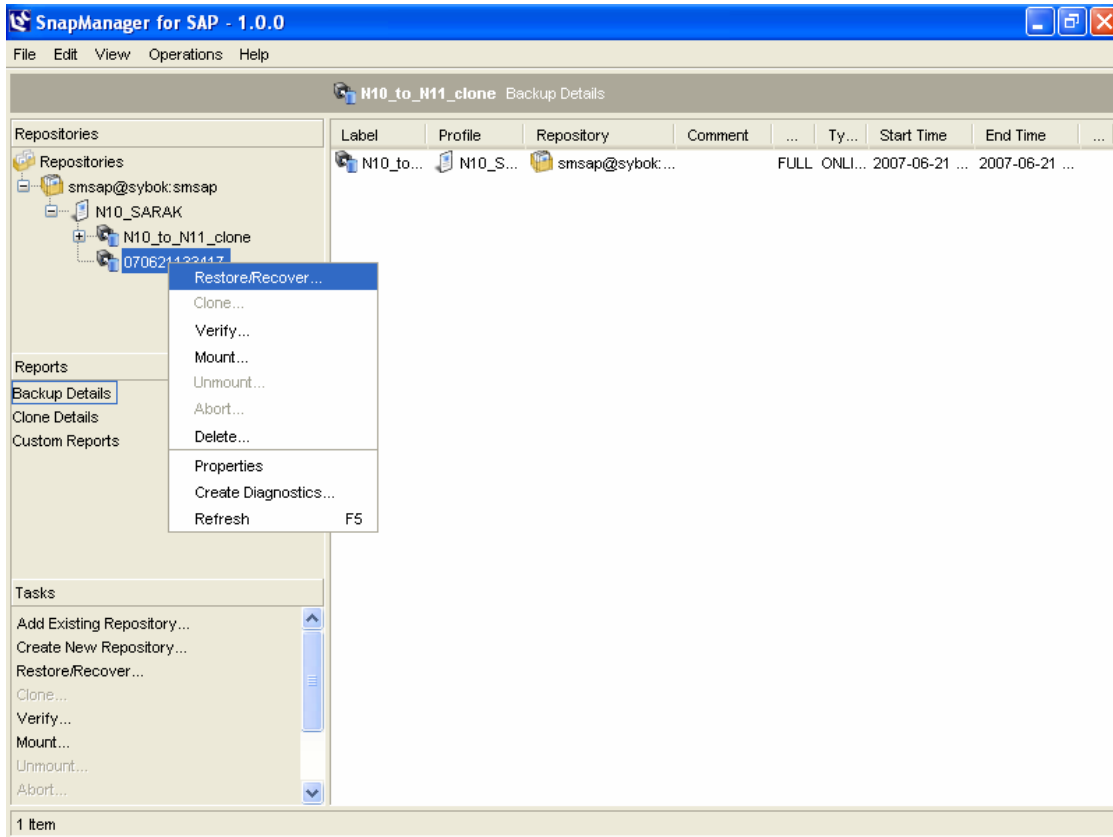
```
brrestore -b bdvlfxx.anf -d util_file -m full
```

In the case that you want to restore only certain datafiles or even a single tablespace, you must specify these options in the BRRESTORE command. This can take quite some time and make the BRRESTORE command very long and cumbersome, which could lead to input errors. It is also necessary to determine the correct command syntax.

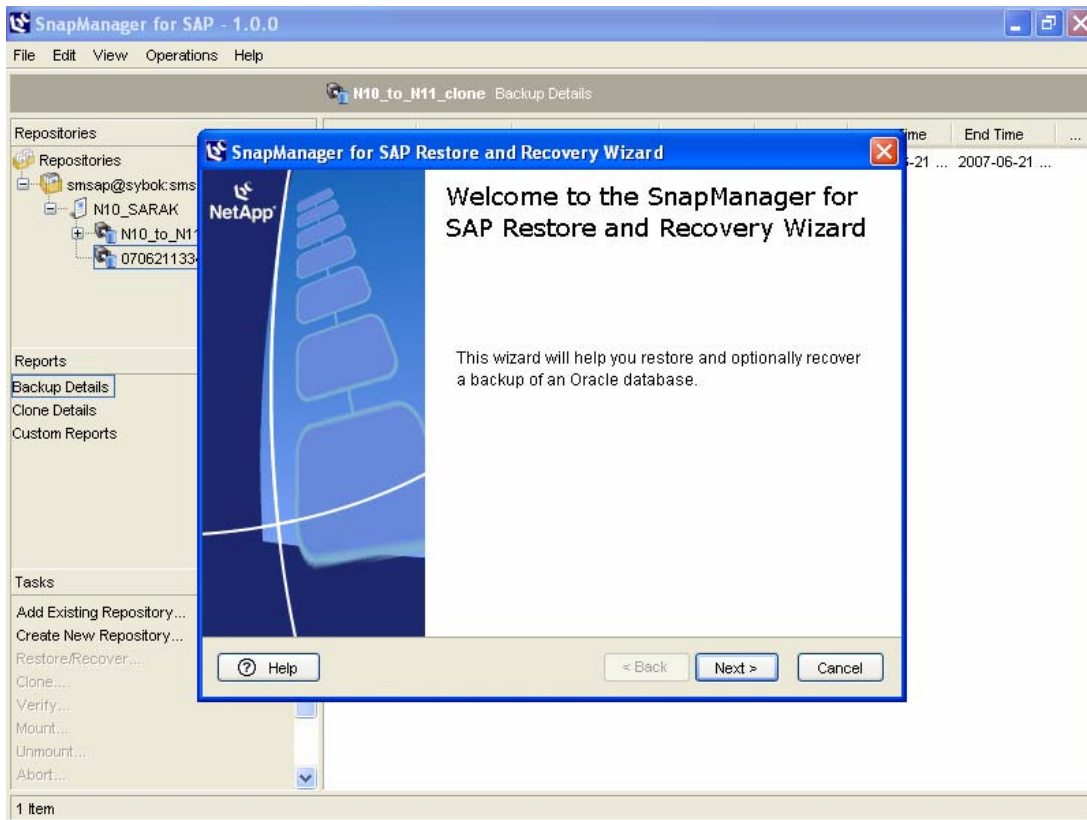
If the backup being restored was an online backup, when the restore has completed it will be necessary to perform a database recovery. A database recovery can be performed using the BR*Tools program BRRECOVER or native database tools.

Restoring backups using the SnapManager for SAP GUI

The process for restoring systems using the SnapManager GUI is straightforward. Start the GUI and open the profile for the system you would like to restore. Select the backup from the list of backups for that system. When you have chosen the backup you'd like to restore, right-click on the backup and choose "Restore/Recovery." A Restore and Recovery wizard will open to guide you through the restoration process. You can also select the backup and then follow the menu path "Operations → Backup → Restore/Recover" and then follow the steps in the Restore and Recovery wizard.



Part of the Restore and Recovery wizard is selecting the point in time for recovering the database. When the restore is completed, SnapManager for SAP will automatically recover the database to the point specified reducing the time necessary to make the system available to the end users.



If only a single datafile or tablespace needs to be restored, this can be easily accomplished using the SnapManager GUI. By choosing the “Selective Restore” option in the Restore and Recovery wizard, you can quickly and easily select the tablespaces or files you wish to restore. These files will then be restored and recovered in one step.

Because of the flexibility, ease of use, and ability to restore and recover the database in a single step provided by the SnapManager GUI, we recommend using the SnapManager for SAP GUI to perform database restore and recovery operations.

4.3 Cloning Systems

System copies are an important part of any SAP environment. Being able to quickly and efficiently produce system copies enables customers to provide high quality SAP solutions while minimizing the risks associated with system upgrades and fixes. NetApp provides the ability to create fast and space-efficient clones using the unique capabilities of Snapshot copies, FlexClone copies, and LUN clones. SnapManager for SAP provides cloning capabilities with an easy-to-use and repeatable process.

Preparing the target system for the clone

Before a clone can be created, the target system needs to be prepared. SnapManager for SAP only clones the Oracle datafiles. As a result, a SAP system has to be installed on the clone target host before a clone can be attached there. You will need to determine the mount points for the cloned file systems and make sure that these mount points are available.

SnapManager will create new configuration files during the clone process. The files from the current installation should be renamed or moved. The files `ORACLE_HOME/dbs/initSID.ora` and `ORACLE_HOME/dbs/orapwSID` should be moved to another location or renamed. If these files still exist during the clone, the clone will fail.

SnapManager will need a location for trace files to be written. Oracle installations for SAP put trace files in `/oracle/<SID>/saptrace/background` and `/oracle/<SID>/saptrace/usertrace`. SnapManager will create a directory in each of those locations based on the cloning specification. Make sure the directory for the trace files does not already exist or the clone will fail.

SnapManager will need a location for log archiving. Oracle installations for SAP use /oracle/<SID>/oraarch for archived log files. SnapManager will create a directory in that filesystem base on the cloning specification. Make sure the directory for the archive logs does not already exist or the clone will fail.

Cloning steps

Once the target system is prepared, the cloning process starts with a backup. Creating a backup using the SnapManager interface is the first step to cloning a system. Only backups created with the SnapManager interface can be used for creating a clone. BRBACKUP backups cannot be used for clones.

The next step is to initiate the cloning process. This is done by right-clicking on the backup to be used for the clone and selecting "Clone" from the menu. The clone wizard is started to guide you through the cloning process. The first time you perform a clone, you will need to proceed through the wizard making the necessary changes to the clone specification.

Some changes that you may find useful that are not part of the default clone specification:

- Change the default mount points to something you prefer
- Create three copies of the control file in the SAP default locations
- Create the same number of redo log files as the SAP default with the same naming conventions and locations
- Change the log archive destination
- Change the Oracle home location
- Change the Oracle userid information

On the "Oracle Home and O/S Account Parameters" screen, there is the potential to save the clone specification in an xml format file. It is recommended that this clone specification be saved after you perform any updates to allow for convenient repeating of a clone from one system to another. Appendix B contains the contents of a clone specification xml file. Saving this file can greatly reduce the amount of time necessary to create a system copy the next time it is required.

Cloning post processing

After the clone has been created, before SAP can be started, it is necessary to complete the post-processing steps. First, the ops\$connect user has to be updated so that the local sidadm and orasid userids can connect to the database. This can be done with a script provided by SAP called ORADBUSR.SQL. This script is available as part of the SAP installation kit and also on the [SAP Service Marketplace](#). The script is an attachment to Note 50088 and can be downloaded directly from the SAP Service Marketplace.

Next, some additional cleanup of the performance tables and other information can be done to prepare the new system for use. Two scripts that can be configured to use after a cloning operation are included in Appendices C, D, and E.

4.4 Maintenance Tasks

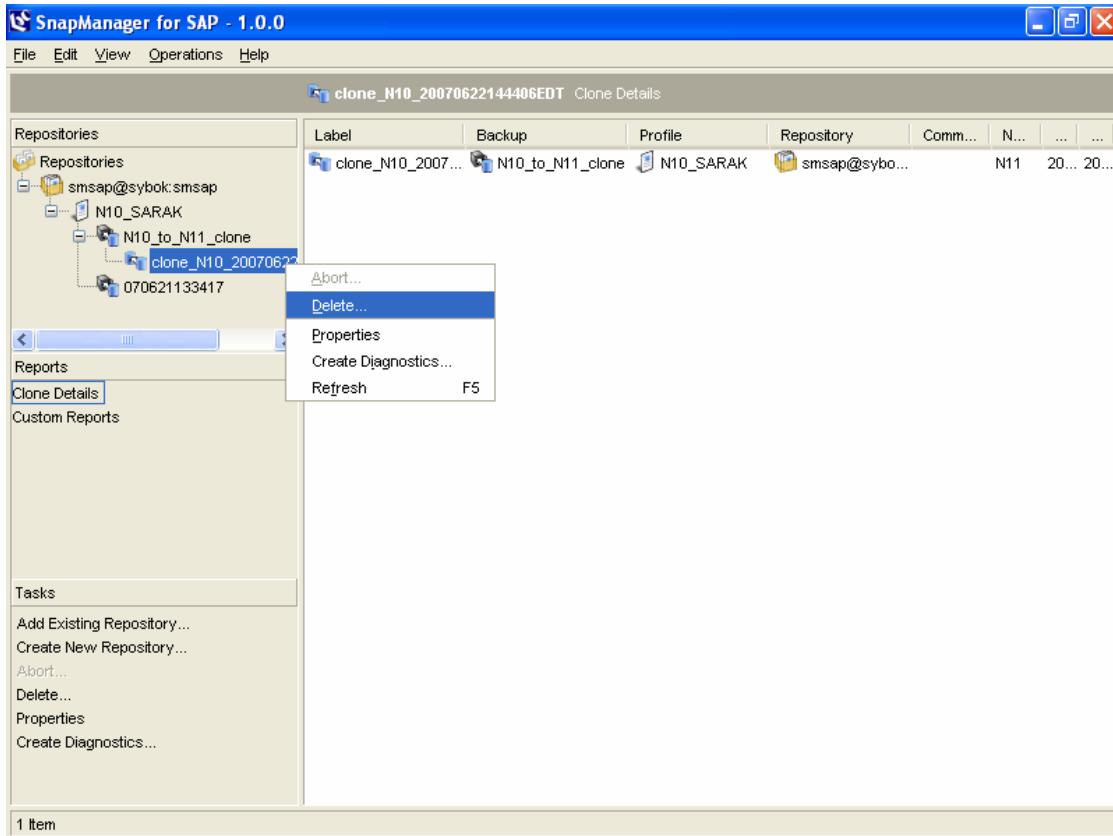
It is important to properly maintain your SnapManager for SAP installation. While the number of Snapshot copies permitted on a volume is large, it is still necessary to periodically clean up old backups. This is especially important for backups created for cloning systems. When these backups are retained for long periods, it may result in a large amount of changed data taking up space in the source volume.

It is important to only maintain SnapManager for SAP Snapshot copies from the SnapManager interface, either GUI or CLI. Deleting backups from the SnapDrive or Data ONTAP interfaces will cause corruption of the SnapManager repository.

Clones

Cloned systems should be deleted and refreshed from a new backup when the size of the Snapshot copy reaches between 10% and 20% of the source system data. For example, a fresh clone may take up only a few megabytes of a 100GB source system. When that clone starts to take up between 10 and 20GB of data, the clone should be deleted and refreshed. This serves to insure that clones do not use too many resources and also guarantees that the clone has the most current data for testing or development.

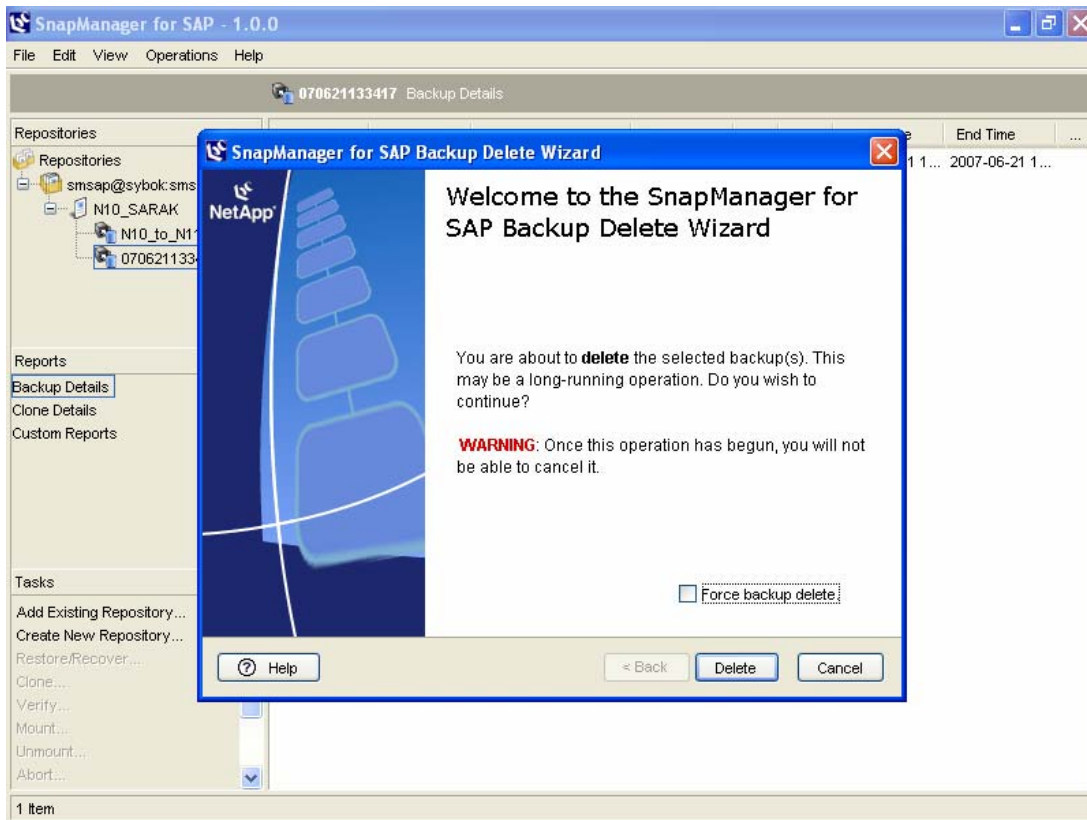
To delete the clone, simply select the clone, right click and choose “Delete.” The SnapManager Clone Delete wizard will lead you through the deletion process. When the clone has been deleted, you can then delete the backup that was used to create the clone. You may need to choose the “Force” option when deleting clones. Review the *SnapManager for SAP v1.0 Installation and Administration Guide* for more information about this option.



Backups

Backups are deleted automatically based on the configuration for the profile for each system. You configure the number of backups to retain when creating the profile. You can change this value anytime by right-clicking on the profile. On the General tab of the properties pop-up that appears, you can change the number of backups to retain for each system. The number of backups to retain is determined by the type of system. For example, you may only keep a single backup of a sandbox system, one week’s worth of backups for a development system, and one month’s worth of backups for a production system. Keep in mind that backups used for creating system copies are included in the number of backups to keep. A backup that is being used for a clone cannot be deleted until the clone is deleted.

To delete the backup, simply select the backup, right-click, and choose “Delete.” The SnapManager Backup Delete wizard will lead you through the deletion process. You may need to choose the “Force” option when deleting backups. Review the *SnapManager for SAP v1.0 Installation and Administration Guide* for more information about this option.



Appendix A – Sample Fibre Channel Storage Setup

This appendix contains commands for creating the filesystems required for a SAP ECC 6.0 system on Solaris™ using SnapDrive for UNIX. To use these commands, it is necessary to first properly configure SnapDrive for UNIX. It is also necessary to create the volumes for the LUNs that will be created.

Note:

- fas3020-1 and fas3020-2 are the storage controllers.
- vol_saplogs_n10 and vol_sapdata_n10 are the volumes on fas3020-1 and fas3020-2 respectively.
- The size of the filesystems will vary for the type of SAP system and release. Consult the SAP installation guide for the correct size of the filesystems.
- There may be additional filesystems required for different SAP systems.
- These commands create the LUNs and disk groups automatically. They also automatically mount the filesystems and create entries in the relevant operating system files to mount the filesystems at startup.

```
# Create the /oracle filesystem
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/oracle.lun01 -fs /oracle -lunsize 110m
```

```
# Create the mount points for the filesystems mounted on /oracle
```

```
mkdir -p /oracle/stage/
mkdir -p /oracle/client/
mkdir -p /oracle/N10
mkdir -p /oracle/N10/102_64
mkdir -p /oracle/N10/origlogA
mkdir -p /oracle/N10/origlogB
mkdir -p /oracle/N10/mirrlogA
mkdir -p /oracle/N10/mirrlogB
mkdir -p /oracle/N10/oraarch
```

```

mkdir -p /oracle/N10/sapreorg
mkdir -p /oracle/N10/sapdata1
mkdir -p /oracle/N10/sapdata2
mkdir -p /oracle/N10/sapdata3
mkdir -p /oracle/N10/sapdata4

# Create the oracle filesystems
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/orastage.lun01 -fs /oracle/stage -lunsize
6600m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/oraclient.lun01 -fs /oracle/client -lunsize
150m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/oraclen10.lun01 -fs /oracle/N10 -lunsize
110m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/orahomen10.lun01 -fs /oracle/N10/102_64
-lunsize 4400m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/origlogan10.lun01 -fs /oracle/N10/origlogA
-lunsize 220m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/origlogbn10.lun01 -fs /oracle/N10/origlogB
-lunsize 220m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/mirrlogan10.lun01 -fs /oracle/N10/mirrlogA
-lunsize 220m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/mirrlogbn10.lun01 -fs /oracle/N10/mirrlogB
-lunsize 220m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/oraarchn10.lun01 -fs /oracle/N10/oraarch -
lunsize 1100m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/sapreorgn10.lun01 -fs
/oracle/N10/sapreorg -lunsize 1700m

# Create the sapdata filesystems
snapdrive storage create -lun fas3020-2:/vol/vol_sapdata_n10/sapdata1n10.lun01 -fs
/oracle/N10/sapdata1 -lunsize 24000m
snapdrive storage create -lun fas3020-2:/vol/vol_sapdata_n10/sapdata2n10.lun01 -fs
/oracle/N10/sapdata2 -lunsize 27000m
snapdrive storage create -lun fas3020-2:/vol/vol_sapdata_n10/sapdata3n10.lun01 -fs
/oracle/N10/sapdata3 -lunsize 24000m
snapdrive storage create -lun fas3020-2:/vol/vol_sapdata_n10/sapdata4n10.lun01 -fs
/oracle/N10/sapdata4 -lunsize 24000m

# Create the sap mount points
mkdir -p /usr/sap/trans
mkdir -p /usr/sap/N10
mkdir -p /sapmnt/N10

# Create the sap filesystems
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/saptransn10.lun01 -fs /usr/sap/trans -
lunsize 2200m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/sapmntn10.lun01 -fs /sapmnt/N10 -lunsize
2200m
snapdrive storage create -lun fas3020-1:/vol/vol_saplogs_n10/ursapn10.lun01 -fs /usr/sap/N10 -lunsize
2200m

```

Appendix B – Sample Cloning Specification – XML Format

Note: This xml file is provided as a sample only and cannot be used in its current format in a customer environment.

```

- <clone-specification xmlns="http://www.netapp.com">
_ <database-specification>

```

```

- <controlfiles>
  <file>/oracle/<TARGET SID>/origlogA/cntrl/cntrl<TARGET SID>.dbf</file>
  <file>/oracle/<TARGET SID>/origlogB/cntrl/cntrl<TARGET SID>.dbf</file>
  <file>/oracle/<TARGET SID>/sapdata1/cntrl/cntrl<TARGET SID>.dbf</file>
  </controlfiles>
- <redologs>
- <redogroup>
  <file>/oracle/<TARGET SID>/mirrlogA/log_g11m2.dbf</file>
  <number>1</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/mirrlogB/log_g12m2.dbf</file>
  <number>2</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/origlogA/log_g13m1.dbf</file>
  <number>3</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/origlogB/log_g12m1.dbf</file>
  <number>5</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/mirrlogB/log_g14m2.dbf</file>
  <number>7</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/origlogB/log_g14m1.dbf</file>
  <number>8</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/origlogA/log_g11m1.dbf</file>
  <number>4</number>
  <size unit="B">52428800</size>
  </redogroup>
- <redogroup>
  <file>/oracle/<TARGET SID>/mirrlogA/log_g13m2.dbf</file>
  <number>6</number>
  <size unit="B">52428800</size>
  </redogroup>
</redologs>
- <parameters>
- <parameter>
  <name>background_dump_dest</name>
  <value>/oracle/<TARGET SID>/saptrace/background/clone</value>
  </parameter>
- <parameter>
  <name>core_dump_dest</name>
  <value>/oracle/<TARGET SID>/saptrace/background/clone</value>
  </parameter>
- <parameter>
  <name>user_dump_dest</name>
  <value>/oracle/<TARGET SID>/saptrace/usertrace/clone</value>
  </parameter>
- <parameter>
  <name>log_archive_dest_1</name>

```

```

<value>LOCATION=/oracle/<TARGET SID>/oraarch/clone</value>
  </parameter>
</parameters>
<oracle-home>/oracle/<TARGET SID>/102_64</oracle-home>
: <oracle-os-account>
  <username>ora<target SID></username>
  <group>dba</group>
  </oracle-os-account>
  <sql-statements />
  </database-specification>
: <storage-specification>
: <storage-mapping>
: <mountpoint>
  <source>/oracle/<SOURCE SID>/sapdata3</source>
  <destination>/oracle/<TARGET SID>/sapdata3clone</destination>
  </mountpoint>
: <mountpoint>
  <source>/oracle/<SOURCE SID>/sapdata2</source>
  <destination>/oracle/<TARGET SID>/sapdata2clone</destination>
  </mountpoint>
: <mountpoint>
  <source>/oracle/<SOURCE SID>/sapdata1</source>
  <destination>/oracle/<TARGET SID>/sapdata1 clone</destination>
  </mountpoint>
: <mountpoint>
  <source>/oracle/<SOURCE SID>/sapdata4</source>
  <destination>/oracle/<TARGET SID>/sapdata4clone</destination>
  </mountpoint>
</storage-mapping>
</storage-specification>
</clone-specification>

```

Appendix C – Cloning Post Processing Script – smsap_clone_post_processing.sh

Note: This script is provided as an example only. This script is not considered a Network Appliance product and is not supported.

```

#!/usr/bin/ksh

# log file writer
#####
write2log()
{
  TEXT=$1
  print -n `date +%Y%m%d%H%M%S` >> $LOG_FILE
  echo "##### $MY_NAME: $TEXT" >> $LOG_FILE
}

# create_osdb_user
#####
create_osdb_user()
{
  ERROR=0
  TMP_LOG="/tmp/tmp_log$$log"

  echo "Creating ops$ user ...."
  echo ""
  write2log "Creating ops$ user ...."
  write2log ""
}

```

```

SQL_SCRIPT="$SYSCOPY_WORKDIR/ORADBUSR.SQL"
CMD="sqlplus /nolog @$SQL_SCRIPT "$SOURCE_SCHEMA_ID" UNIX "$TARGET_SAPSID" x"
su - ora"$LOW_TARGET_DBSID" -c "$CMD" > $TMP_LOG

cat $TMP_LOG | grep "unable" > /dev/null 2>&1
if [ $? -gt 0 ]
then
  cat $TMP_LOG | grep "error" > /dev/null 2>&1
  if [ $? -gt 0 ]
  then
    cat $TMP_LOG | grep "cannot" > /dev/null 2>&1
    if [ $? -gt 0 ]
    then
      ERROR=0
    else
      ERROR=1
    fi
  else
    ERROR=1
  fi
else
  ERROR=1
fi
if [ $ERROR -gt 0 ]
then
  echo "Creating ops$ failed !"
  write2log "Creating ops$ failed !"
fi
cat $TMP_LOG >> $LOG_FILE
rm $TMP_LOG
}

# insert_sapuser_pw
#####
insert_sapuser_pw()
{
  ERROR=0
  TMP_LOG="/tmp/tmp_log$$log"
  SAPUSER_PW_SCRIPT="$SYSCOPY_WORKDIR/"$TARGET_DBSID"/insert_sapuser_pw.sql"

  echo "Updating "$LOW_TARGET_SAPSID"adm.sapuser table and setting new password .... "
  echo ""
  write2log "Updating "$LOW_TARGET_SAPSID"adm.sapuser table and setting new password .... "
  write2log ""

  echo "connect / as sysdba" > $SAPUSER_PW_SCRIPT
  echo "alter user "$SOURCE_SCHEMA_ID" identified by "$SAPUSER_PW";" >>
  $SAPUSER_PW_SCRIPT
  echo "insert into ops\$"$LOW_TARGET_SAPSID"adm.sapuser values
  ("'$SOURCE_SCHEMA_ID';"'$SAPUSER_PW');" >> $SAPUSER_PW_SCRIPT
  echo "commit;" >> $SAPUSER_PW_SCRIPT
  echo "exit" >> $SAPUSER_PW_SCRIPT

  CMD="sqlplus /nolog @$SAPUSER_PW_SCRIPT"
  su - ora"$LOW_TARGET_DBSID" -c "$CMD" > $TMP_LOG

  cat $TMP_LOG | grep "1 row created." > /dev/null 2>&1
  if [ $? -eq 0 ]
  then
    cat $TMP_LOG | grep "User altered." > /dev/null 2>&1
    if [ $? -eq 0 ]

```

```

then
    ERROR=0
else
    ERROR=1
    echo "Setting password of "$LOW_TARGET_SAPSID"adm failed .... "
    write2log "Setting password of "$LOW_TARGET_SAPSID"adm failed .... "
fi
else
    ERROR=1
    echo "Updating "$LOW_TARGET_SAPSID"adm.sapuser table failed .... "
    write2log "Updating "$LOW_TARGET_SAPSID"adm.sapuser table failed .... "
fi
cat $TMP_LOG >> $LOG_FILE
rm $TMP_LOG
}

```

```

# add file to temp tablespace
#####
add_temp_file()
{
    ERROR=0
    TMP_LOG="/tmp/tmp_log$$log"

    echo "Adding file to Temp tablespace....."
    echo ""
    write2log "Adding file to Temp tablespace....."
    write2log ""

    CMD="sqlplus /nolog @""$SYSCOPY_WORKDIR"/add_temp_file.sql"
    su - ora"$LOW_TARGET_DBSID" -c "$CMD" > $TMP_LOG
    cat $TMP_LOG | grep "Tablespace altered." > /dev/null 2>&1
    if [ $? -eq 0 ]
    then
        ERROR=0
    else
        ERROR=1
    fi
    if [ $ERROR -gt 0 ]
    then
        echo "Adding file to Temp tablespace failed"
        write2log "Adding file to Temp tablespace failed"
    fi
    cat $TMP_LOG >> $LOG_FILE
    rm $TMP_LOG
}

```

```

# do_sap_post_processing
#####
do_sap_post_processing()
{
    ERROR=0
    TMP_LOG="/tmp/tmp_log$$log"

    SQL_TEMPLATE=""$SYSCOPY_WORKDIR"/sap_post_processing.tpl"

    SQL_SCRIPT=""$SYSCOPY_WORKDIR"/"$TARGET_DBSID"/sap_post_processing_"$TARGET_
    SAPSID".sql"

```

```

echo "Updating some SAP tables .... "
echo ""
write2log "Updating some SAP tables .... "
write2log ""

cat $$SQL_TEMPLATE | sed "s/owner/${SOURCE_SCHEMA_ID}/g" > $$SQL_SCRIPT
CMD="sqlplus /nolog @$SQL_SCRIPT"
su - ora"$LOW_TARGET_DBSID" -c "$CMD" > $TMP_LOG
cat $TMP_LOG | grep "ERROR" > /dev/null 2>&1
if [ $? -gt 0 ]
then
    cat $TMP_LOG | grep "ORA-" > /dev/null 2>&1
    if [ $? -gt 0 ]
    then
        ERROR=0
    else
        ERROR=1
    fi
else
    ERROR=1
fi
if [ $ERROR -gt 0 ]
then
    echo "Updating SAP tables failed !"
    echo ""
    write2log "Updating SAP tables failed !"
    write2log ""
fi
cat $TMP_LOG >> $LOG_FILE
rm $TMP_LOG
}

#####
# main
#####

SYSCOPY_WORKDIR=/oracle/<SID>/smsap
SOURCE_SCHEMA_ID=<SCHEMA>
TARGET_SAPSID=<SID>
LOW_TARGET_DBSID=<sid>
LOW_TARGET_SAPSID=<sid>
TMP_LOG=/<directory>/tmp_log$$log
LOG_FILE=/<directory>/clone$$log
SAPUSER_PW=sap

create_osdb_user

insert_sapuser_pw

add_temp_file

do_sap_post_processing

```

Appendix D – Clone Post Processing Template – insert_sapuser_pw.sql

Note: This script is provided as an example only. This script is not considered a Network Appliance product and is not supported.

```
connect / as sysdba
```



```
alter user SAPSR3 identified by sap;
insert into ops$<sid>adm.sapuser values ('SAPSR3','sap');
commit;
exit
```

Appendix E – Clone Post Processing Template – smsap_post_processing.tpl

Note: This script is provided as an example only. This script is not considered a Network Appliance product and is not supported.

```
-- Template file for SnapManager for SAP clone post activities
-- =====
-- "owner" is a placeholder for the schema owner and will be
-- automatically adapted to "sapr3" or sap<schema-id> when the
-- sql script is executed by smsap_clone_post_processing.sh
--
-- Additional lines and operation can be added to the file to
-- adapt to the customer environment

connect / as sysdba
-- delete performance statistics
delete from owner.sdbah;
delete from owner.sdbad;
delete from owner.sdbap;
delete from owner.sdbar;
delete from owner.osmon;
delete from owner.pahi;
delete from owner.moni;
delete from owner.dbsnp;
commit;

-- delete default, start and instance profile
delete from owner.tpfet;
delete from owner.tpfht;
delete from owner.tpfid;
commit;

-- delete table used for buffer synchronization
delete from owner.ddlog;
commit;

-- delete logon groups
delete from owner.rzllitab;
commit;

-- delete database statistics
delete from owner.dbstathora;
delete from owner.dbstatiora;
delete from owner.dbstattora;
delete from owner.dbstaihora;
commit;

-- delete jobs
delete from owner.tbtc0
where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SAP_%' and
```

```
jobname not like 'COLLECTOR_FOR%' and
jobname not like 'SYST%';
commit;
delete from owner.tbtc
where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SAP_%' and
jobname not like 'COLLECTOR_FOR%' and
jobname not like 'SYST%';
commit;
delete from owner.tbtcp
where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SAP_%' and
jobname not like 'COLLECTOR_FOR%' and
jobname not like 'SYST%';
commit;
delete from owner.btcevtjob
where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SAP_%' and
jobname not like 'COLLECTOR_FOR%' and
jobname not like 'SYST%';
commit;

exit
```

Appendix F – Temp Tablespace File Add – add_temp_file.sql

Note: This script is provided as an example only. This script is not considered a Network Appliance product and is not supported.

```
CONNECT / AS SYSDBA;
ALTER TABLESPACE PSAPTEMP ADD TEMPFILE '/oracle/<SID>/sapdata2clone/temp_1/temp.data1'
    SIZE 2000M REUSE AUTOEXTEND ON NEXT 20971520 MAXSIZE 10000M;
exit
```

Additional References

SAP and NetApp Best Practices

[SAP on UNIX and Oracle with FCP and Network Appliance Storage](#)

[SAP on UNIX and Oracle with NFS and Network Appliance Storage](#)

Databases and Operating System Tuning

[Database Layout with Data ONTAP 7G](#)

[Network Appliance Best Practices for Oracle](#)

NetApp Support

[SnapManager for SAP Product Interoperability Matrix](#)

[NetApp Knowledge Base Articles](#)

SAP

[SAP Service Marketplace](#)

[SAP Developer Network](#)

[SAP Help Portal](#)

Revision History

Date	Name	Description
8/29/07	Craig Sullivan	Final revisions
8/22/2007	Craig Sullivan	Review editorial comments
05/15/2007	Craig Sullivan	Creation

